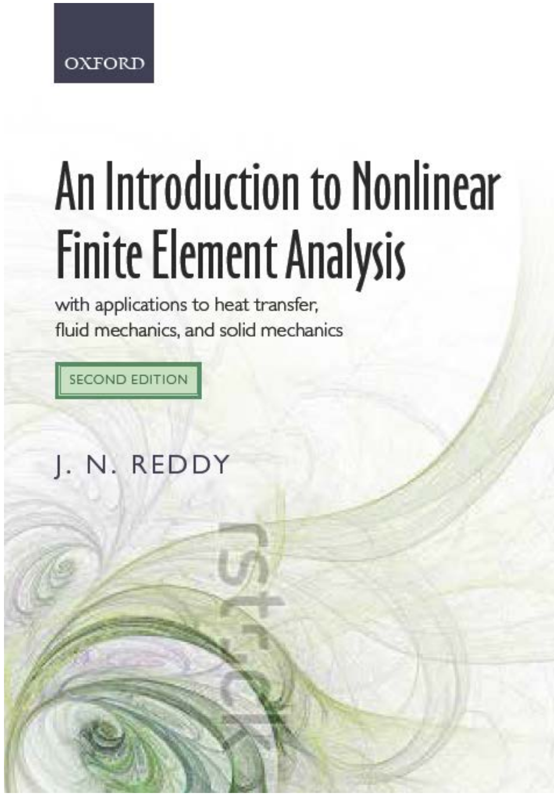
The Finite Element Method

2D Nonlinear Finite Element Analysis

Read: Chapter 6



CONTENTS

- Weak Form development
- Finite element model
- Computation of tangent coefficients
- Review of numerical integration
- Computer implementation
- Summary

Finite Element Formulation of a Model 2D Nonlinear Equation

Model Equation

$$-\frac{\partial}{\partial x}\left(a_{11}\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(a_{22}\frac{\partial u}{\partial y}\right)-f=0 \qquad (1)$$

$$a_{11}=a_{11}(x,y,u,u_{,x},u_{,y}),\quad a_{22}=a_{22}(x,y,u,u_{,x},u_{,y})$$

Weak Form

$$0=\int_{\Omega^e}\left[\frac{\partial w_i}{\partial x}\left(a_{11}\frac{\partial u_h}{\partial x}\right)+\frac{\partial w_i}{\partial y}\left(a_{22}\frac{\partial u_h}{\partial y}\right)-w_i f\right]dx-\oint_{\Gamma^e} w_i\, q_n\, ds \quad (2)$$

Primary and Secondary Variables

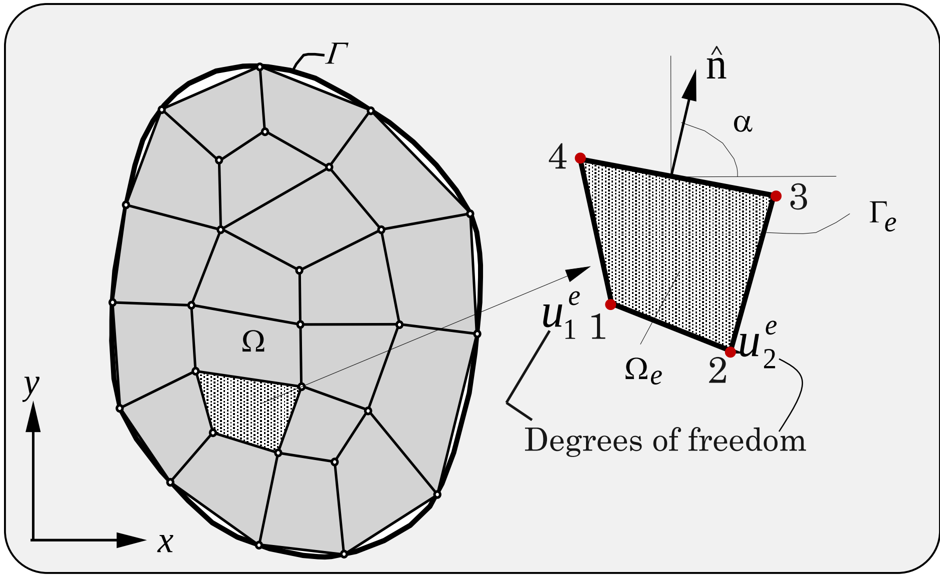
$$\text{PV: } u_h \qquad \text{SV: } q_n=\left(a_{11}\frac{\partial u_h}{\partial x}\right)n_x+\left(a_{22}\frac{\partial u_h}{\partial y}\right)n_y$$

FINITE ELEMENT FORMULATION ...(continued)

Finite Element Approximation

$$u(x,y) \approx u_h^e(x,y) = \sum_{j=1}^{n} u_j^e \psi_j^e(x,y)$$

Finite Element Model



$$0 = \sum_{j=1}^{n} u_j \int_{\Omega^e} \left(a_{11} \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} + a_{22}\frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial y} \right) dxdy - \int_{\Omega^e} \psi_i f\, dxdy - \oint_{\Gamma^e} \psi_i\, q_n\, ds$$

$$= \sum_{j=1}^{n} K_{ij}^e u_j^e - f_i^e - Q_i^e = \sum_{j=1}^{n} K_{ij}^e u_j^e - F_i^e$$

$$K_{ij}^e = \int_{\Omega^e} \left(a_{11}(u) \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} + a_{22}(u)\frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial y} \right) dxdy$$

$$F_i^e = \int_{\Omega^e} \psi_i f\, dxdy + \oint_{\Gamma^e} \psi_i\, q_n\, ds$$

The coefficient matrix $[K]$ is a function of the nodal values of u

COMPUTATION OF TANGENT COEFFICIENTS

$$R_i = \sum_{j=1}^{n} K_{ij}^e u_j^e - F_i^e \quad \text{or} \quad \{R^e\} = [K^e(u^e)]\{u^e\} - \{F^e\}$$

$$T_{ij}^e = \frac{\partial R_i}{\partial u_j} = \sum_{p=1}^{n} \left(\frac{\partial K_{ip}^e}{\partial u_j} u_p^e + K_{ip}^e \frac{\partial u_p}{\partial u_j} \right) = \sum_{p=1}^{n} \frac{\partial K_{ip}^e}{\partial u_j} u_p^e + K_{ij}^e$$

Suppose that $a_{11} = a_{11}^0 + a_{11}^1 u$ and $a_{22} = a_{22}^0 + a_{22}^1 u$. Then

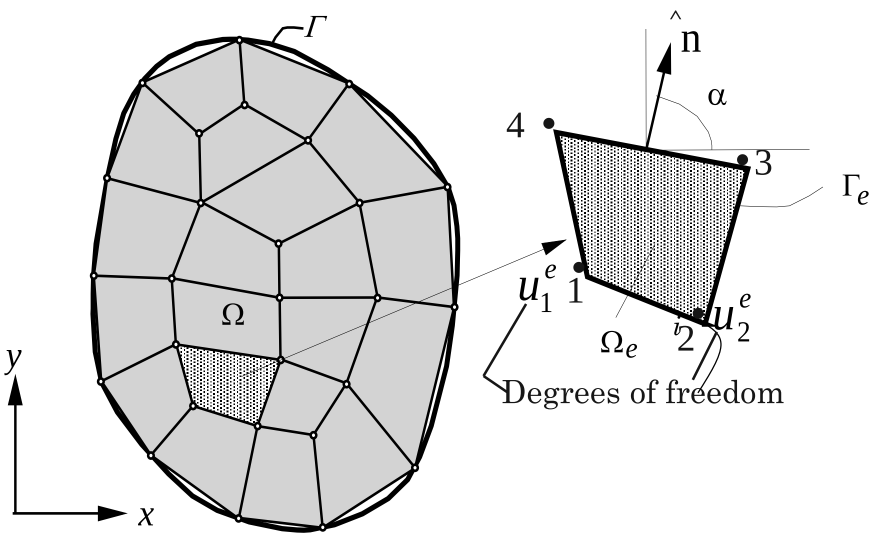
$$T_{ij}^e = K_{ij}^e + \sum_{p=1}^{n} u_p^e \int_{\Omega^e} \left(\frac{\partial a_{11}}{\partial u_j} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_p}{\partial x} + \frac{\partial a_{22}}{\partial u_j} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_p}{\partial y} \right) dxdy$$

$$\frac{\partial a_{11}}{\partial u_j} = a_{11}^1 \frac{\partial u}{\partial u_j} = a_{11}^1 \frac{\partial}{\partial u_j} \sum_{p=1}^{n} u_p \psi_p = a_{11}^1 \psi_j, \quad \frac{\partial a_{22}}{\partial u_j} = a_{22}^1 \frac{\partial u}{\partial u_j} = a_{22}^1 \psi_j,$$

$$T_{ij}^e = K_{ij}^e + \int_{\Omega^e} \left(a_{11}^1 \frac{\partial u}{\partial x} \frac{\partial \psi_i}{\partial x} + a_{22}^1 \frac{\partial u}{\partial y} \frac{\partial \psi_i}{\partial y} \right) \psi_j \, dxdy$$

NUMERICAL EVALUATION OF INTEGRAL COEFFICIENTS

- **Transformation of the integrals posed on arbitrary-shaped element to the master element domain so that evaluation of the integrals is made easy.**
- **The Gauss integration rule that evaluates an integral expression as a linear sum of the integrand evaluated at certain points (Gauss points) and weights (Gauss weights) is used.**



$$u(x,y) \approx u_h^e(x,y) = \sum_{j=1}^{n} u_j^e \psi_j^e(x,y)$$

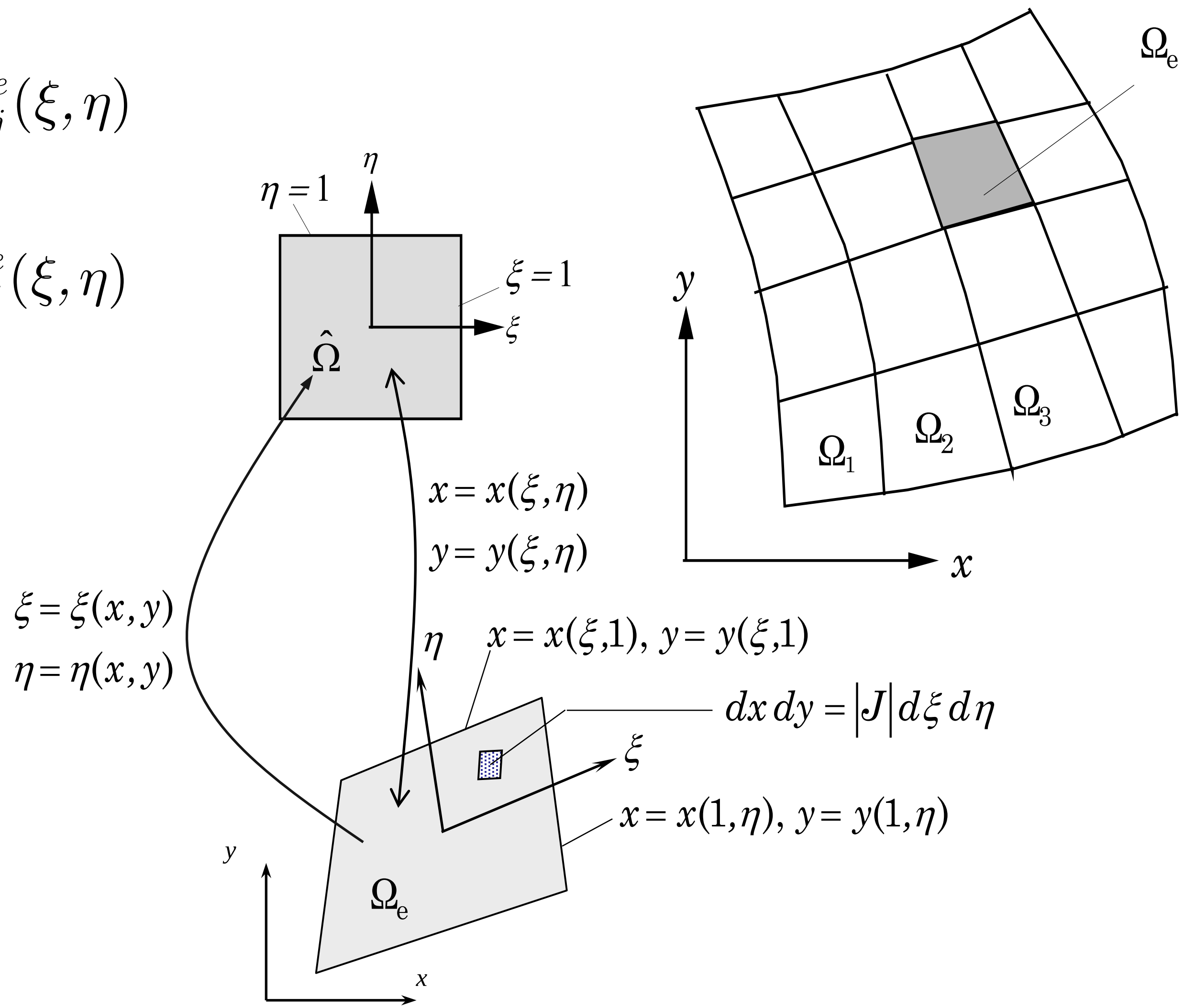
$$\approx u_h^e(x(\xi,\eta), y(\xi,\eta))$$

$$= \sum_{j=1}^{n} u_j^e \psi_j^e(\xi,\eta)$$

NUMERICAL INTEGRATION

$$x(\xi,\eta) = \sum_{j=1}^{m} x_j^e \hat{\psi}_j^e(\xi,\eta)$$

$$y(\xi,\eta) = \sum_{j=1}^{m} y_j^e \hat{\psi}_j^e(\xi,\eta)$$



NUMERICAL EVALUATION OF INTEGRALS

$$K_{ij}^{e} = \int_{\Omega_e} \left[a_{11} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + a_{22} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right] dxdy$$

$$= \int_{\Omega_e} F_{ij}(x,y)\, dxdy = \int_{\hat{\Omega}} F_{ij}(x(\xi,\eta), y(\xi,\eta))\, J\, d\xi d\eta$$

$$= \int_{\hat{\Omega}} F_{ij}(\xi,\eta)\, J\, d\xi d\eta \approx \sum_{I=1}^{NGP} \sum_{J=1}^{NGP} W_I\ W_J\ \hat{F}_{ij}(\xi_I, \eta_J)$$

Using the chain-rule, we obtain

$$\frac{\partial \psi_i}{\partial \xi} = \frac{\partial \psi_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial \psi_i}{\partial y}\frac{\partial y}{\partial \xi}$$

$$\frac{\partial \psi_i}{\partial \eta} = \frac{\partial \psi_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial \psi_i}{\partial y}\frac{\partial y}{\partial \eta}$$

$$\Rightarrow \begin{Bmatrix} \dfrac{\partial \psi_i}{\partial \xi} \\ \dfrac{\partial \psi_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \psi_i}{\partial x} \\ \dfrac{\partial \psi_i}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \dfrac{\partial \psi_i}{\partial x} \\ \dfrac{\partial \psi_i}{\partial y} \end{Bmatrix}$$

NUMERICAL INTEGRATION implementation point of view

Jacobian matrix

$$[J]=\begin{bmatrix}\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi}\\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta}\end{bmatrix}=\begin{bmatrix}\sum_{i=1}^{m} x_i\frac{\partial \hat{\psi}_i}{\partial \xi} & \sum_{i=1}^{m} y_i\frac{\partial \hat{\psi}_i}{\partial \xi}\\ \sum_{i=1}^{m} x_i\frac{\partial \hat{\psi}_i}{\partial \eta} & \sum_{i=1}^{m} y_i\frac{\partial \hat{\psi}_i}{\partial \eta}\end{bmatrix}$$

$$=\begin{bmatrix}\frac{\partial \hat{\psi}_1}{\partial \xi} & \frac{\partial \hat{\psi}_2}{\partial \xi} & \dots & \frac{\partial \hat{\psi}_m}{\partial \xi}\\ \frac{\partial \hat{\psi}_1}{\partial \eta} & \frac{\partial \hat{\psi}_2}{\partial \eta} & \dots & \frac{\partial \hat{\psi}_m}{\partial \eta}\end{bmatrix}\begin{bmatrix}x_1 & y_1\\ x_2 & y_2\\ \vdots & \vdots\\ x_m & y_m\end{bmatrix},\quad dxdy = J\,d\xi\,d\eta$$

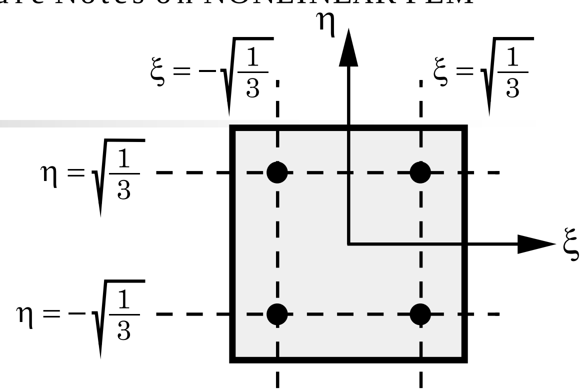
Global derivatives in terms of the local derivatives

$$\begin{Bmatrix}\frac{\partial \psi_i}{\partial x}\\ \frac{\partial \psi_i}{\partial y}\end{Bmatrix}=[J]^{-1}\begin{Bmatrix}\frac{\partial \psi_i}{\partial \xi}\\ \frac{\partial \psi_i}{\partial \eta}\end{Bmatrix}=[J^*]\begin{Bmatrix}\frac{\partial \psi_i}{\partial \xi}\\ \frac{\partial \psi_i}{\partial \eta}\end{Bmatrix}$$

ELEMENT CALCULATIONS

$$
\begin{aligned}
K_{ij}^{e} &= \int_{\Omega_e} \left[a_{11} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + a_{22} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right] dxdy \\
&= \int_{\hat{\Omega}} \left\{ a_{11}(\xi,\eta) \left(J_{11}^{*} \frac{\partial \psi_i}{\partial \xi} + J_{12}^{*} \frac{\partial \psi_i}{\partial \eta} \right) \left(J_{11}^{*} \frac{\partial \psi_j}{\partial \xi} + J_{12}^{*} \frac{\partial \psi_j}{\partial \eta} \right) \right. \\
&\qquad \left. + a_{22}(\xi,\eta) \left(J_{21}^{*} \frac{\partial \psi_i}{\partial \xi} + J_{22}^{*} \frac{\partial \psi_i}{\partial \eta} \right) \left(J_{21}^{*} \frac{\partial \psi_j}{\partial \xi} + J_{22}^{*} \frac{\partial \psi_j}{\partial \eta} \right) \right\} J \, d\xi d\eta \\
&= \int_{\hat{\Omega}} \hat{F}_{ij}^{e}(\xi,\eta)\, d\xi\, d\eta = \int_{-1}^{1} \int_{-1}^{1} \hat{F}_{ij}^{e}(\xi,\eta)\, d\xi\, d\eta \\
&\approx \sum_{I=1}^{NGP_\xi} \sum_{J=1}^{NGP_\eta} \hat{F}_{ij}^{e}(\xi_I, \eta_J) W_I W_J
\end{aligned}
$$

GAUSS QUADRATURE



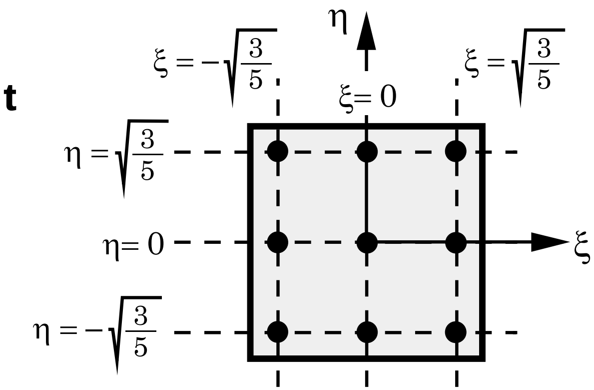
$$\int_{\Omega_e} F_{ij}(x,y)\,dxdy = \int_{\hat{\Omega}_e} \hat{F}_{ij}(\xi,\eta)\; d\xi\, d\eta$$

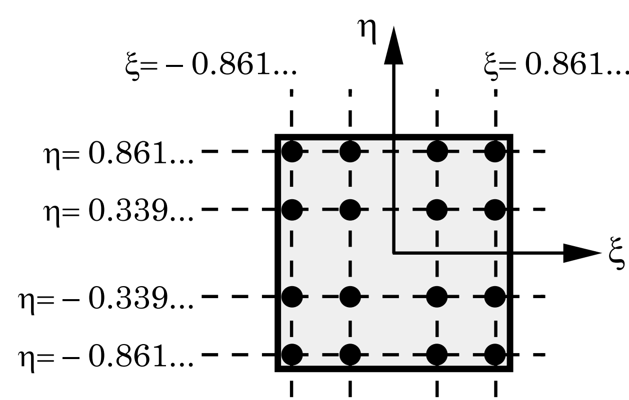
$$\approx \sum_{I,J=1}^{N} W_I W_J \hat{F}_{ij}(\xi_I,\eta_J)$$

Domain of the *master* element

Domain of the *physical* element

Gauss *points* **Gauss *weights***





COMPUTER IMPLEMENTATION (2D)

Variables used in the program, FEM2D

Read the input data description for FEM2D to know the meaning of all input variables; other variables are described below.

NPE - nodes per element, n

ELXY(i, j) - Global coordinates of the ith node of element e, (x_i^e, y_i^e)

ELK(i, j) - Element coefficient, K_{ij}^e

ELF(i) - Element coefficient, f_i^e

ELU(i) - Element solution, u_i^e

TAN(i, j) - Element coefficient, $\left(K_{ij}^e\right)^{\tan}$

$AX0, AX1, AU$ – Coefficients in the definition of $a(x)$:

$$a(x) = AX0 + AX1 * x + AU * u$$

COMPUTER IMPLEMENTATION (2D)

SFL(i) – Element shape (or approximation) funtion, ψ_i^e

DSFL(i, j) – Derivative of the ith shape function with respect to the local (normalized) coordinates ξ and η

$$(j=1):\ \frac{\partial \psi_i}{\partial \xi},\ (j=2):\ \frac{\partial \psi_i}{\partial \eta}$$

GDSFL(i, j) – Derivative of the ith shape function with respect to the global coordinates x and y: $\frac{\partial \psi_i}{\partial x}$ and $\frac{\partial \psi_i}{\partial y}$

A11 = A10 + A1X*X + A1Y*Y; A22 = A20 + A2X*X + A2Y*Y; A00=CONST

IEL = 1, Linear rectangular element

IEL= 2, Quadratic rectangular element

GLXY(I,1) – Global x coordinate of the Ith global node

GLXY(I,2) – Global y coordinate of the Ith global node

NEQ – Number of equations in the mesh

COMPUTER IMPLEMENTATION (continued)

Variables used in the program, FEM2D (continued)

$$SF(i) = \psi_i,\ \ XI = \xi,\ \ ETA = \eta,\ \ DET = J,\ \ DSF(1,i) = \frac{\partial \psi_i}{\partial \xi},\ \ DSF(2,i) = \frac{\partial \psi_i}{\partial \eta}$$

$$GDSF(1,i) = \frac{\partial \psi_i}{\partial x},\ \ GDSF(2,i) = \frac{\partial \psi_i}{\partial y}$$

$$ELU(i) = u_i,\ \ ELF(i) = f_i,\ \ ELK(i,j) = K_{ij},\ \ ELXY(i,1) = x_i,\ \ ELXY(i,2) = y_i$$

$$K_{ij}^e = \int_{\Omega_e} \left[a_{11} \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} + a_{22} \frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial y} \right] dxdy = \int_{\hat{\Omega}} F_{ij}(\xi,\eta)\, J\, d\xi d\eta$$

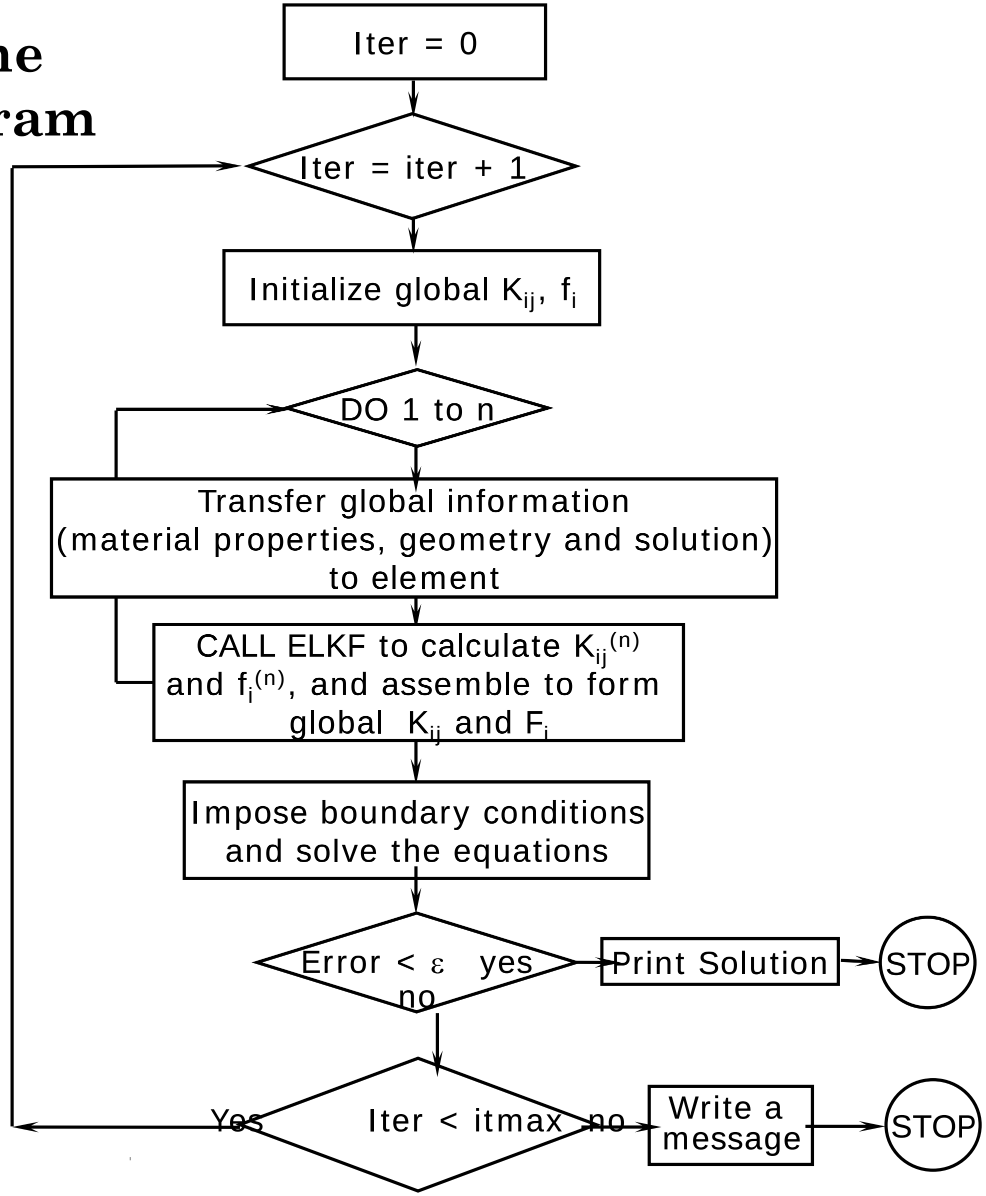
$$\approx \sum_{I=1}^{NGP} \sum_{J=1}^{NGP} \underbrace{W_I\ W_J\ J}_{\text{CONST}}\ F_{ij}(\xi_I,\eta_J)$$

$$F_{ij}(\xi,\eta) = a_{11}(\xi,\eta) \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} + a_{22}(\xi,\eta) \frac{\partial \psi_i}{\partial y}\frac{\partial \psi_j}{\partial y}$$

$$F(i,j) = \left[A11 * GDSF(1,i) * GDSF(1,j) + A22 * GDSF(2,i) * GDSF(2,j)\right]$$

GENERAL LOGIC FOR THE NONLINEAR ANALYSIS

Logic in the MAIN program



COMPUTER IMPLEMENTATION for the nonlinear analysis

Logic in the MAIN program

```
      ITER=0
160 ITER=ITER+1
      IF(ITER.GT.ITMAX)THEN
         WRITE(IT,1200)ITMAX
         STOP
      ENDIF
C     Initialize the global coefficient matrices and vectors
      DO 180 I=1,NEQ
      GLF(I)=0.0
      DO 180 J=1,NHBW
180 GLK(I,J)=0.0
C     Compute element matrices and assemble the matrices
      DO 250 N=1,NEM
      DO 200 I=1,NPE
      NI=NOD(N,I)
      IF(NONLN.GT.0)THEN
         ELU(I)=GLU(NI)
      ENDIF
      ELXY(I,1)=GLXY(NI,1)
      ELXY(I,2)=GLXY(NI,2)
200 CONTINUE
       ......
250 CONTINUE
```

COMPUTER IMPLEMENTATION for the nonlinear analysis (continued)

```
C     Save the previous iteration solution and update the current one
      IF(NONLN.GT.0)THEN
        DO 270 I=1,NEQ
        GPU(I)=GLU(I)
        IF(NONLN.EQ.1)THEN
          GLU(I)=GLF(I)
        ELSE
          GLU(I)=GLU(I)+GLF(I)
        ENDIF
  270  CONTINUE
C     Test for the convergence of the solution
        IF(NONLN.GT.0)THEN
          DNORM=0.0
          DINORM=0.0
          DO 280 I=1,NEQ
          DNORM=DNORM+GLU(I)*GLU(I)
          IF(NONLN.EQ.1)THEN
            DINORM=DINORM+(GLU(I)-GPU(I))**2
          ELSE
            DINORM=DINORM+GLF(I)*GLF(I)
          ENDIF
  280     CONTINUE
```

```
          ERROR=DSQRT(DINORM/DNORM)
            IF(ERROR.GT.EPS)GOTO 160
              DO I=1,NEQ
              GLF(I)=GLU(I)
              ENDDO
              WRITE(ITT,1100)ITER,ERROR
            ENDIF
          ENDIF
C     Print the solution (i.e., nodal values of
the primary variables)
```

Logic in the MAIN program

COMPUTER IMPLEMENTATION for the nonlinear analysis (continued)

Logic in the ELMATRCS2D subroutine

```
      SUBROUTINE ELMATRCS2D (NPE,NN,INTF,NONLN,ITYPE)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/STF/ELF(9),ELK(9,9),ELXY(9,2),ELU(9)
      COMMON/PST/A10,A1X,A1Y,A20,A2X,A2Y,A00,F0,FX,FY,
     *                           A1U,A1UX,A1UY,A2U,A2UX,A2UY
      COMMON/SHP/SF(9),GDSF(2,9)
      DIMENSION  GAUSPT(5,5),GAUSWT(5,5),TANG(9,9)
      COMMON/IO/IN,ITT
C
      DATA GAUSPT/5*0.0D0, -0.57735027D0, 0.57735027D0, 3*0.0D0,
     2  -0.77459667D0, 0.0D0, 0.77459667D0, 2*0.0D0, -0.86113631D0,
     3  -0.33998104D0, 0.33998104D0, 0.86113631D0, 0.0D0, -0.90617984D0,
     4  -0.53846931D0,0.0D0,0.53846931D0,0.90617984D0/
C
      DATA GAUSWT/2.0D0, 4*0.0D0, 2*1.0D0, 3*0.0D0, 0.55555555D0,
     2   0.88888888D0, 0.55555555D0, 2*0.0D0, 0.34785485D0,
     3 2*0.65214515D0, 0.34785485D0, 0.0D0, 0.23692688D0,
     4   0.47862867D0, 0.56888888D0, 0.47862867D0, 0.23692688D0/
```

COMPUTER IMPLEMENTATION for the nonlinear analysis (continued)

Logic in the ELMATRCS2D subroutine (continued)

```
C
C      Initialize the arrays
C
      DO 100 I = 1,NPE
             ELF(I)  = 0.0
      DO 100 J = 1,NPE
             IF(ITYPE.GT.1)THEN
             TANG(I,J)=0.0
             ENDIF
  100       ELK(I,J)= 0.0
C
C      Do-loops on numerical
C      integration begin here.
C     Subroutine INTERPLN2D
C     is called here
C
```

```
      DO 200 NI = 1,IPDF
      DO 200 NJ = 1,IPDF
         XI  = GAUSPT(NI,IPDF)
         ETA = GAUSPT(NJ,IPDF)
         CALL INTERPLN2D (NPE,XI,ETA,DET,ELXY)
         CNST = DET*GAUSWT(NI,IPDF)*GAUSWT(NJ,IPDF)
         X=0.0
         Y=0.0
         U=0.0
         UX=0.0
         UY=0.0
         DO 140 I=1,NPE
            U=U+ELU(I)*SF(I)
            UX=UX+ELU(I)*GDSF(1,I)
            UY=UY+ELU(I)*GDSF(2,I)
            X=X+ELXY(I,1)*SF(I)
140        Y=Y+ELXY(I,2)*SF(I)
```

COMPUTER IMPLEMENTATION for the nonlinear analysis (continued)

Logic in the ELMATRCS2D subroutine (continued)

```
C   Define the coefficients of the
C   differential equation
        FXY=F0+FX*X+FY*Y
        A11=A10+A1X*X+A1Y*Y
        A22=A20+A2X*X+A2Y*Y
        IF(NONLN.GT.0)THEN
A11=A11+A1U*U+A1UX*UX
*                +A1UY*UY
A22=A22+A2U*U+A2UX*UX
*                +A2UY*UY
        ENDIF
C  Define the element source vector
C   and coefficient matrix
       DO 180 I=1,NPE
ELF(I)=ELF(I)+FXY*SF(I)*CNST
```

```
         DO 180 I=1,NPE
           ELF(I)=ELF(I)+FXY*SF(I)*CNST
           DO 160 J=1,NPE
           S00=SF(I)*SF(J)*CNST
           S11=GDSF(1,I)*GDSF(1,J)*CNST
           S22=GDSF(2,I)*GDSF(2,J)*CNST
           ELK(I,J)=ELK(I,J)+A11*S11+A22*S22+A00*S00
C     Define the part needed to add to [K] to define [T]
           IF(ITYPE.GT.1)THEN
             S10=GDSF(1,I)*SF(J)*CNST
             S20=GDSF(2,I)*SF(J)*CNST
             S12=GDSF(1,I)*GDSF(2,J)*CNST
             S21=GDSF(2,I)*GDSF(1,J)*CNST
             TANG(I,J)=TANG(I,J)
  *              +UX*(A1U*S10+A1UX*S11+A1UY*S12)
  *              +UY*(A2U*S20+A2UX*S21+A2UY*S22)
           ENDIF
160        CONTINUE
180     CONTINUE
200 CONTINUE
```

COMPUTER IMPLEMENTATION for the nonlinear analysis (continued)

Logic in the ELMATRCS2D subroutine (continued - end)

$$K_{ij}^{\tan} = \frac{\partial R_i^e}{\partial u_j^e} = K_{ij}^e + \sum_{p=1}^{n} \frac{\partial K_{ip}^e}{\partial u_j^e} u_p^e$$

$$[K^{\mathrm{Tan}}(\{U\}^r)]\{\delta U\} = \{F\}^r - [K(U^r)]^r\{U\}^r$$

```
C     Compute the residual vector and tangent matrix
       IF(ITYPE.GT.1)THEN
          DO 220 I=1,NPE
          DO 220 J=1,NPE
  220         ELF(I)=ELF(I)-ELK(I,J)*ELU(J)
          DO 240 I=1,NPE
          DO 240 J=1,NPE
  240         ELK(I,J)=ELK(I,J)+TANG(I,J)
          ENDIF
C
          RETURN
          END
```

SUMMARY

(1) Finite element formulation of a two-dimensional model nonlinear equation in a single variable
(2) Computation of tangent matrix coefficients
(3) Numerical integration of matrix coefficients
(4) Computer implementation of 2D problem

All other items (e.g., iterative methods for the solution of nonlinear equations, error check, and computer logic for nonlinear analysis) remain the same as in the 1-D problems discussed earlier.